UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/008,691	11/08/2001	Jay P. Hoeflinger	042390.P11920	7462

8791 7590 08/19/2005

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
KISS, ERIC B	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 08/19/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/008,691	HOEFLINGER ET AL.	
	Examiner	Art Unit	
	Eric B. Kiss	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 June 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. The reply filed 9 June 2005 has been received and entered. Claims 1-26 are pending.

Response to Amendment

2. Applicant's amendment to claim 21 appropriately addresses the objection to this claim based on an informality. Accordingly, this objection is withdrawn in view of Applicant's amendments.

3. Applicant's amendment to claim 23 appropriately addresses the rejection of this claim under 35 U.S.C. §112, second paragraph, based on indefiniteness. Accordingly, this rejection is withdrawn in view of Applicant's amendment.

Response to Arguments

4. Applicant's arguments filed 9 June 2005 have been fully considered but they are not persuasive.

a. In response to Applicant's arguments regarding claims 1-3 and 11-13, the Examiner submits that the variable having a TARGET attribute meets the recited "descriptor pointing to a target data". As is well known in the art, a variable is a named storage location, with a variable's name being used as a descriptor referring to (pointing to) one or more memory locations.

b. In response to Applicant's arguments regarding claims 4-10 and 14-26, the Examiner submits that, while the C_LOC(x) function achieves interoperability between a Fortran program and a C program, the C_LOC(x) function is actually a Fortran-language function, and not a C-language (or another, "different from Fortran"-language) function as Applicant contends.

Specification

5. In the previous Office action, the Examiner provided guidelines from MPEP §608.01(v) in regard to Applicant's use of various trademarks throughout the specification. In particular, the previous Office action stated,

> Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and **every effort made** to prevent their use in any manner which might adversely affect their validity as trademarks. [emphasis added]

Further, the Examiner provided several suggestions as to how Applicant may amend the specification in order to use the affected trademarks in a more appropriate manner.

It is noted that the Examiner indicated that Applicant was "free to apply or ignore these suggestions at their discretion," and Applicant has deliberately chosen to ignore the Examiner's suggestions. However, it should be further noted that the Examiner's statements do not serve to relieve Applicant of their responsibility to respect the intellectual property of others, and Applicant is strongly encouraged to reconsider the Examiner's comments on the use of trademarks in the previous Office action.

As the guidelines for appropriate usage of trademarks within the specification (MPEP §608.01(v)) are set forth in permissive language (using the word "should" instead of, for example, "shall" or "must"), any potential repercussions of Applicant's discretionary actions are

apparently beyond the scope of the instant prosecution. Accordingly, the Examiner will not pursue this issue further (unless the facts at hand change significantly enough to warrant further action).

Claim Rejections - 35 USC § 103

6. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

7. Claims 1-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Working Draft: J3/00-007R3," October 2000 (hereinafter *Fortran2000*) in view of Alfred V. Aho, et al., "Compilers: Principles, Techniques, and Tools," 1988 (hereinafter *Aho et al.*).

As per claim 1, *Fortran2000* discloses generating a function having an argument, the function expressed in a high-level programming language, wherein the function includes a set of one or more instructions to return a memory address of the argument as a result of the function (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396); and generating a call to the function, the call expressed in the high-level programming language, wherein the call passes a descriptor as the argument, the descriptor pointing to a target data (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396).

Fortran2000 is intentionally silent on the mechanism by which programs are transformed for use on computing systems (see section 1.4 on p. 1).

However, *Aho et al.* teaches the known use of a compiler unit to arrive at a machine-executable implementation of program source code (see, for example, "The Context of a Compiler" on page 4, along with Fig. 1.3 on page 5; see further, page 2, third paragraph, describing, very briefly, an early Fortran compiler). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to use such a compiler unit to process the source code disclosed by *Fortran2000* in order to arrive at a working implementation of the source code through known means.

As per claim 2, *Fortran2000* further discloses the high-level programming language including a Fortran programming language (the entire *Fortran2000* document is part of a Fortran programming language specification; see, for example, section 1.1 on page 1). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 3, *Fortran2000* further discloses the argument being any available type, including an integer type, provided it has the TARGET attribute (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 4, *Fortran2000* discloses receiving a first code, wherein the first code refers to a variable of a target data type, wherein the variable is addressable using a descriptor (see, for example, section 16.2 on pages 392-401); and translating the first code into a second code, the second code expressed in a high-level programming language (see, for example, the description

of *C_LOC (X)* in section 16.2.3 on pages 395-396), wherein the translation requires a memory address of the descriptor, and wherein the translation comprises: generating a function having an argument, wherein the function is expressed in the high level programming language, and the function includes a set of one or more instructions that return the memory address of the argument as a result of the function (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396); and generating a call to the function, wherein the call passes the descriptor as the argument (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396).

Fortran2000 is intentionally silent on the mechanism by which programs are transformed for use on computing systems (see section 1.4 on p. 1).

However, *Aho et al.* teaches the known use of a compiler unit to arrive at a machine-executable implementation of program source code (see, for example, "The Context of a Compiler" on page 4, along with Fig. 1.3 on page 5; see further, page 2, third paragraph, describing, very briefly, an early Fortran compiler). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to use such a compiler unit to process the source code disclosed by *Fortran2000* in order to arrive at a working implementation of the source code through known means.

As per claim 5, *Fortran2000* discloses the translating comprising generating an interface block for the function for each different target data type in the first code (see, for example,

section 16.2 on pages 392-401). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 6, *Fortran2000* further discloses the high-level programming language including a Fortran programming language (the entire *Fortran2000* document is part of a Fortran programming language specification; see, for example, section 1.1 on page 1). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 7, *Fortran2000* further discloses the argument being any available type, including an integer type, provided it has the TARGET attribute (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 8, *Fortran2000* further discloses generating a data structure to store information based on the target data type (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396; the *C_LOC (X)* function produces a *C_PTR* scalar return value). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claims 9 and 10, *Fortran2000* further discloses the function including a routine from a runtime library written in a C programming language, the routine to return a memory address of an argument of the routine (see, for example, section 16.2 on pages 392-401, and in

particular, see the description of *C_LOC (X)* in section 16.2.3 on pages 395-396). Therefore, for reasons stated above, such claims also would have been obvious.

As per claims 11-13, these are machine-readable medium versions of the claims methods discussed above (claims 1-3, respectfully). Further, the use of such a machine-readable medium is inherent in implementing the computer software methods disclosed in *Fortran2000*. Therefore, for reasons stated above, such claims also would have been obvious.

As per claims 14-20, these are machine-readable medium versions of the claims methods discussed above (claims 4-10, respectfully). Further, the use of such a machine-readable medium is inherent in implementing the computer software methods disclosed in *Fortran2000*. Therefore, for reasons stated above, such claims also would have been obvious.

As per claim 21, *Fortran2000* discloses a translation unit to receive a first code that refers to a variable of a target data type, wherein the variable is referred to by a descriptor (see, for example, section 16.2 on pages 392-401), the translation unit to translate the first code into a second code, the second code based on a high-level programming language (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396), wherein the translation requires a memory address of the descriptor and wherein the translation comprises: generating a function having an argument, wherein the function is expressed in the high level programming language, and the function includes a set of one or more instructions to return the memory address of the argument as a result of the function (see, for example, the description of *C_LOC*

(X) in section 16.2.3 on pages 395-396); and generating a call to the function, wherein the call passes the descriptor as the argument (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396);

Fortran2000 is intentionally silent on the mechanism by which programs are transformed for use on computing systems (see section 1.4 on p. 1).

However, *Aho et al.* teaches the known use of a compiler unit and a linker unit to arrive at a machine-executable implementation of program source code (see, for example, "The Context of a Compiler" on page 4, along with Fig. 1.3 on page 5; see further, page 2, third paragraph, describing, very briefly, an early Fortran compiler). Therefore, it would have been obvious to one of ordinary skill in the computer art at the time the invention was made to use such a compiler unit and a linker unit to process the source code disclosed by *Fortran2000* in order to arrive at a working implementation of the source code through known means.

As per claim 22, *Fortran2000* further discloses the generation of the second code including the generation of a function, the function having an entity as an argument, and a call to the function, wherein the call to the function accepts the argument as an entity for which the memory address can be determined and returned as a result of the function (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396). The use of a compiler unit to provide the requisite functionality has been addressed as set forth above for claim 21. Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 23, *Fortran2000* further discloses the entity being any available type, including an integer type, provided it has the TARGET attribute (see, for example, the description of *C_LOC (X)* in section 16.2.3 on pages 395-396). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claim 24, *Fortran2000* further discloses the high-level programming language including a Fortran programming language (the entire *Fortran2000* document is part of a Fortran programming language specification; see, for example, section 1.1 on page 1). Therefore, for reasons stated above, such a claim also would have been obvious.

As per claims 25 and 26, *Fortran2000* further discloses the function including a routine from a runtime library written in a C programming language, the routine to return a memory address of an argument of the routine (see, for example, section 16.2 on pages 392-401, and in particular, see the description of *C_LOC (X)* in section 16.2.3 on pages 395-396).

Conclusion

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

9. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Eric B. Kiss whose telephone number is (571) 272-3699. The Examiner can normally be reached on Tue. - Fri., 7:00 am - 4:30 pm. The Examiner can also be reached on alternate Mondays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tuan Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry of a general nature should be directed to the TC 2100 Group receptionist: 571-272-2100.

EBK/EBK
August 9, 2005

TUAN DAM
SUPERVISORY PATENT EXAMINER